(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 335 283 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
13.08.2003 Bulletin 2003/33

(51) Int Cl.[7]: G06F 9/445

(21) Application number: 03290180.3

(22) Date of filing: 23.01.2003

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT SE SI SK TR
Designated Extension States:
AL LT LV MK RO

(30) Priority: 24.01.2002 CA 2369228

(71) Applicant: Alcatel Canada Inc.
Kanata, Ontario K2K 2E6 (CA)

(72) Inventors:
- Swanson, Sheldon Keith John
Ottawa, Ontario K1H 7K7 (CA)
- Macqueen, Kenneth Glenn
Merrickville, Ontario K0G 1N0 (CA)

(74) Representative: Feray, Valérie et al
Feray Lenne Conseil
44/52, Rue de la Justice
75020 Paris (FR)

(54) System and method for managing configurable elements of devices in a network element

(57) The invention provides a system and a method of selecting a version of a program code from a series of program codes for a program element associated with a hardware element for a circuit board. The invention identifies a compatible program code for a particular hardware element and downloads the compatible program code. A record is associated with the hardware element to identify its compatible program codes. The program codes are stored in a bundle and are associated with a software code. A pogram loader, associated with each bundle, checks the compatibilities of the hardware element to both the software code and the selected program code. Another record stores compatibility information of the program codes with hardware element versions and is also associated with the bundle. Code downloads are made first to an inactive memory bank and after the codes are verified, the inactive memory bank is made active.

BEST AVAILABLE COPY

EP 1 335 283 A2

Description

[0001] The invention relates to a system and method for managing the tracking of configurable devices in an electronic device.

[0002] It is common for electronic and communication devices to utilize hardware, software and firmware elements to provide aspects of their functionality. The hardware elements may comprise upgradable components such as CPUs, memory modules and other upgradeable and replaceable devices. Firmware stored in non-volatile electronic devices, such as EEPROMs, EPROMs, PLDs and PGAs, and volatile devices such as FPGAs and RAM, provide low level, executable code performing various calculations and operations on input and output signals depending on the program embodied in the firmware. Software, which is frequently stored non-volatile memory, but executes from volatile memory such as RAM, provides an operating code for a program which is executed on a CPU.

[0003] After an electric device is tested and released into a product any one of the hardware, software or firmware elements may be independently upgraded on a device. Upgrades may be implemented to an element to correct existing operational errors or to improve its performance. For example, in hardware a "cut and strap" may be added to change an electrical connection on one of its circuits. In software, a new routine may be added providing a new feature for the program. As upgrades are implemented to an element, they are tracked using release codes. A series of release codes document the upgrades history of the element. For a given electronic device, the elements and their associated release codes determine what versions of the hardware, software and firmware are present in the device.

[0004] It is possible that for any one element, an upgraded version may not operate with the existing elements on the device thereby requiring upgrades to those elements. For example, a hardware upgrade implemented to a CPU may require that the software and/or the firmware be upgraded. Alternatively, it is possible that the upgraded version of the element may be backwards compatible with the other existing elements, but may not be able to perform its upgraded features without an accompanying upgrade to one of the other elements. For example, an upgraded CPU providing additional D/A conversion facilities may work with the existing firmware and software on a device, but access to the D/A facilities would not be provided without an upgrade to the software. It will be appreciated that other permutations of forward and backward compatibility levels are possible for hardware, software and firmware elements of an electronic device.

[0005] Accordingly, one object of the present invention is to provide method of selecting a version of a program code from a series of program codes for a program element associated with a hardware element for a circuit board that allows tracking the compatibility of various versions of hardware, software and firmware associated with an electronic device or a system of electronic devices.

[0006] More precisely, the present invention provides a method of selecting a version of a program code from a plurality of program codes for a program storage element associated with a hardware element for a circuit board, said hardware element being implementable in a plurality of hardware element versions, said method comprising:

- identifying a particular program code of said plurality of program codes which is compatible with a particular hardware element of said plurality of hardware element versions on said circuit board;
- selecting from a group of program codes selected from said plurality of program codes a selected program code matching said particular program code; and
- loading said program storage element with said selected program code.

[0007] Thus, the method according to the invention identifies a compatible program code for a particular hardware element and downloads the compatible program code. A record is associated with the hardware element to identify its compatible program codes. The program codes are for example stored in a bundle and are associated with a software code. A program loader, associated with each bundle, checks the compatibilities of the hardware element to both the software code and the selected program code. The method according to the invention is a centralized method that allows for tracking independently compatibilities for firmware and software elements which are affected or affect the operation of hardware elements in electronic circuits.

[0008] The method may have information regarding the particular program code stored with the circuit board, the program codes stored in a program code bundle and the selected program code is identified by accessing a list. The list has an entry for each of the hardware element versions, with each entry having a reference to a compatible program code which can operate with the entry.

[0009] The method may have each program code being a firmware program.

[0010] The method may have a software program being associated with the program code bundle and being associated with the hardware element.

[0011] The method may have each entry in the list further associated with a list of earlier versions of each program code with which the hardware element is compatible.

[0012] The method may have each entry in the list further associated with a list of earlier versions of the software program with which the hardware element is compatible.

[0013] In a second aspect, a system for processing

BEST AVAILABLE COPY

program updates associated with a hardware element is provided. The hardware element can be implemented in several versions. The hardware element is associated with a circuit board. The system comprises a record of the hardware element versions, each entry in the record being associated with a compatible program code from a set of program codes, a second record associated with the circuit board, the second record identifying a particular program code which is compatible with an installed hardware element version of the hardware element on the circuit board, a bundle of executeable program codes from a group of the program codes and a program code loader. The loader can access the record and the second record to identify a suitable program code from the plurality of program codes for the installed hardware element version, locate the suitable program code in the bundle, and effect transfer of the suitable program code from the bundle to a storage device. The storage device can receive the suitable code and is associated with the installed hardware element version and the circuit board.

[0014] The system may have the program code as being a firmware program.

[0015] The system may have a software program associated with the program code bundle and with the hardware element.

[0016] The system may have each the entry in the record being further associated with a compatible software program.

[0017] The system may have each entry in the record being further associated with a list of earlier versions of each program element with which the hardware element is compatible.

[0018] The system may have each entry in the record being further associated with a list of earlier versions of the software program with which the hardware element is compatible.

[0019] In a third aspect, a method for updating a program code from a series of program codes for a hardware element is provided for a circuit board. The hardware element may be implemented in a series of versions. The method comprises identifying a particular program code which is compatible with a particular hardware element on the circuit board, selecting from a group of program codes a selected program code matching the particular program code, and transferring the suitable program code from the group of program codes to a storage device. The storage device can receive the suitable code and is associated with the particular hardware element and the circuit board.

[0020] The method may further utilize information regarding the particular program code is stored with the circuit board, may have the program codes stored in a program code bundle, and may have the selected program code identified by accessing a list. The list may comprise an entry for each hardware element version, with each entry in the list being associated with a reference to a compatible program code which can operate with the entry.

[0021] The method may have each of the program codes being a firmware program.

[0022] The method may have a software program associated with the program code bundle and associated with the hardware element.

[0023] The method may have each entry is the list being further associated with a list of earlier versions of each program code with which the hardware element is compatible.

[0024] The method may have each entry in the list being further associated with a list of earlier versions of the software program with which the hardware element is compatible.

[0025] In a fourth aspect, a method of selecting a version of a firmware code for a programmable firmware element associated with a hardware element for a circuit board is provided. The hardware element may be implemented in a series of versions; the hardware element may further be associated with a software code. The method comprises identifying a particular version of the hardware element versions which is associated with the circuit board, identifying a particular firmware code which is compatible with a particular hardware element on the circuit board, selecting from a group of firmware codes a selected firmware code matching the particular firmware code, associating a software code with the group of firmware codes, loading the firmware element with the selected firmware code, and checking compatibility of operation of the software code with the selected firmware code and the particular version of the plurality of hardware element versions. In doing so, the method select the selected firmware code independently of installation of the software code.

[0026] In a fifth aspect, a method of evaluating compatibility of a hardware element in a circuit card with a program code is provided. The hardware element may be implemented in a series of versions. The program code may be implemented in a series of versions. The method comprises identifying a particular hardware element version which is associated with the circuit board, identifying a particular program which is associated with the hardware element, evaluating whether the particular program code is compatible with the particular hardware element version and indicating whether the particular program code is compatible with the particular hardware element version.

[0027] The method may have information regarding the particular program code identified by accessing a list. The list may have an entry for each of the hardware element version. Each entry in the list may have a reference to a compatible program code which can operate therewith.

[0028] The method may have each program code being a firmware code.

[0029] In other aspects of the invention, various combinations and subset of the above aspects are provided.

[0030] The foregoing and other aspects of the inven-

BEST AVAILABLE COPY

tion will become more apparent from the following description of specific embodiments thereof and the accompanying drawings which illustrate, by way of example only, the principles of the invention. In the drawings, where like elements feature like reference numerals (and wherein individual elements bear unique alphabetical suffixes):

Fig. 1 shows a block diagram of hardware elements of a circuit card of an embodiment of the invention;

Fig. 2 shows a table providing a compatibility matrix of software and firmware codes associated with hardware elements of the circuit card of Fig. 1;

Fig. 3 shows a block diagram of hardware elements of a main controller card communicating with modules via a bus of another embodiment of the invention;

Fig. 4 shows a block diagram of a resource module associated with the main controller card of the embodiment shown in Fig. 3; and

Fig. 5 shows a block diagram a network of modules communicating with a resource module of yet another embodiment of the invention.

[0031] The description which follows, and the embodiments described therein, are provided by way of illustration of an example, or examples, of particular embodiments of the principles of the present invention. These examples are provided for the purposes of explanation, and not limitation, of those principles and of the invention. In the description which follows, like parts are marked throughout the specification and the drawings with the same respective reference numerals.

Basic Features of System

[0032] The following is a description of an electronic system associated with an embodiment.

[0033] Referring to Fig. 1, electronic circuit card 100 is shown. Circuit card 100 has various electronic modules which operate to provide manipulation and production of various electronic signals, including main controller 102 and modules 104a, 104b and 104c. Controller 102 communicates with modules 104 through electronic connections 106 in card 100. The interface point between controller 102 and connections 106 is shown as node 108. Similarly, the interface between connections 106 and each of modules 104 is shown as nodes 110. Other devices 112 may also be connected either directly or indirectly to connections 106 thereby allowing them to communicate with either controller 102 or modules 104. As shown, an additional set of connections 114 is provided from main controller 102 to an external connector 116, thereby allowing circuit card 100 to be associated with other external devices (not shown). Each of connections 106 and 114 may comprise n separate electrical tracks for carrying signals between controller 102 and modules 104.

[0034] Controller 102 provides the main controlling system for modules 104. On controller card 102 there is CPU 118, RAM 120, firmware devices 122 and non-volatile memory 124. It will be appreciated that there may be several of any of the elements on main controller 102. For the shown implementation, RAM 120 stores an operating program which is executed on CPU 118. Firmware devices 122 are programmable, each having separate program codes. Non-volatile memory 124 is also programmable and stores configuration information about controller card 102. Non-volatile memory device 124 can be implemented as a Serial Electrically Erasable PROM (SEEP) and is used to store local configuration information relating to the hardware, software and firmware elements in module 104. It acts as a virtual label for the contents of the module 104. Further detail on the contents of SEEP 124 are provided below. Internal connections 126 provide an electrical connection between the elements of controller 102 and interface node 108.

[0035] Generally, each of modules 104 comprises the following hardware, software and firmware elements. Hardware element 128, abstractly, is an upgradable component, and is shown as local CPU 130. In an embodiment, CPU 130 is mounted directly to module 104 in order to dissuade switch operators from making their own modifications to module 104. However in another embodiment, CPU 130 is mounted in a socket 132 which provides a releasable interface for the electrical contact points of CPU 130 to other components in modules 104. Socket 132 may comprise a zero-insertion force ("ZIF") socket, known in the art. Accordingly, with socket 132, CPU 130 may be removed and replaced with an upgraded component. Also, an upgrade on hardware element 128 may be a physical modification, such as a "cut and strap" made to electronic tracks on the module, shown as modification 140. The software element 134 is an executable program code stored in a volatile program storage element, or memory device, such as RAM, and is controlled by an aspect of the hardware element 128. For the example, the software element is stored in local RAM 136 and operates on CPU 130. Software, as is known, frequently undergoes revisions and has to be installed into a released circuit card 100. The firmware elements 138 are executeable program codes and are provided in a number of program storage elements, including programmable volatile and non-volatile devices such as PLDs, FPGAs and other devices 142. Each of the firmware elements 138 may utilize different program codes. As with software elements, for firmware elements 138 often are upgraded, which occasionally have to be installed into an existing circuit card 100. An elec-

BEST AVAILABLE COPY

tronic signature containing information on the hardware elements 128 on module 104 is stored in SEEP 144. Further detail on the use of the electronic signature and the configuration information in SEEPs 124 and 144 are provided later. It will be appreciated that other modules 104 may not have all of the hardware, software and firmware elements.

[0036] As described earlier, it is possible that for card 100, an upgraded hardware element 128 may not be compatible with the installed software elements 134 and firmware elements 138. Card 100 may not be able to implement the features of the upgraded hardware element 128 without other upgrades to other software and firmware elements 138. Alternatively, the upgraded hardware element 128 may cause card 100 to malfunction without other upgrades. Alternatively still, an upgraded hardware, software or firmware element may be fully compatible with the existing other elements.

[0037] In the embodiment, a set of compatible firmware elements 138 which can operate with a particular version of a software element 134 are bundled together into program bundle, referred to as an application bundle 146. The application bundle is stored in a flash bank of memory 148. In the embodiment, there is an active flash bank 148 and an inactive flash bank 148. The operating firmware and software elements are stored in the active flash bank 148 and new downloads are stored in the inactive flash bank 148. Once the new download is verified, the embodiment can switch between the active and inactive flash banks 148. Accordingly; when the particular software element 134 is provided to main module 104, the embodiment has the ability of selecting the most appropriate set of firmware elements 138 for that particular software element 134. In the embodiment, each software element 134 has a record, which is accessible by other elements (such as a firmware loader, described later) of firmware elements 138 and hardware elements 128 which will operate with it.

[0038] Referring to Fig. 2, the embodiment provides a system and method for tracking compatible software elements 134 and firmware elements 138 for a particular hardware element 128 on card 100. Matrix 200 is used to track this information. The matrix 200 tracks hardware elements 128 of card 100, which, when they are upgraded, require a change to the existing firmware elements 138 and/or software elements 134 and it provides for a hardware element 128 a record of compatible firmware elements 138 and software elements 134 for each upgrade. Field 200A provides a marker for the version of the matrix stored for a given set of hardware elements 128. For the example, the version of matrix 200 is "Card 100 Rel 1". Matrix 200 is stored at a centrally accessible location, such as in flash memory 148 on main controller 102. The data format of matrix 200 may comprise any machine readable format, including an ASCII file, a spreadsheet file or a coded binary file. Accordingly, main controller 102 has access to data indicating the hardware/firmware/software element compatibilities for itself and its associated modules 104.

[0039] In matrix 200, a row entry is provided for a particular hardware element 128. Fields in the row provide information about its current set of compatible firmware elements 138 and software elements 134. The row also contains information on the compatibility of firmware and software elements with each other. In describing the rows and columns of matrix 200, first a description is provided of the relationship between a hardware element 128 and its affected firmware elements 138. Next, a description is provided of the relationship between a hardware element 128 and its affected software elements 134. Finally, a description is provided for the relationship between a firmware element 138 and its affected software elements 134.

[0040] In matrix 200, for a hardware element 128 listed in a row, entries in column 202 of the row identify hardware elements 128 for the module 104 which affect firmware elements 138 and software elements 134. As an example, entries 202A, 202B and 202C relate to a first CPU, such as CPU 130A, on card 100 "CPU A", versions "Rel 1" and "Rel 2"; entry 202D related to a second CPU, such as CPU 130B, "CPU B", having "Rel 1"; entry 202E relates to the card 100 itself, as "cut and strap" revisions may be made to the card itself such as modification 140.

[0041] For the hardware/firmware element relationships, for each hardware element 128 identified in row 202, a list of firmware elements 138 whose ability to function are affected by the upgrade to that hardware element 128 is provided in entries in the corresponding set of columns 204. Each entry in set of columns 204 is a reference number and version number for a particular firmware download for the affected firmware element 138. Accordingly for the example in set of columns 204, three firmware downloads are identified relating to the first CPU. They are identified as "FirmElem1 Rel1" in column 204A(1), "FirmElem2 Rel1" in column 204A(2) and "FirmElem3 Rel1" in column 204A(3); each firmware download is the latest version of each firmware element which is compatible with that upgraded version of that hardware element 128 in entry 202A.

[0042] Entries in matrix 200 indicate the compatibility levels for each upgradeable hardware element 128 vis-à-vis firmware elements 138. A Hardware/Firmware Compatibility Level (HFCL) entry is provided for each upgraded hardware element 128 in matrix 200. The entry in the HFCL field 206 tracks a release level of sets of firmware elements 138 (listed in sets of columns 204) which are compatible with that version of the hardware element 128. In the embodiment, an integer is used as a value to track release levels. A base hardware element 128 is provided with a HFCL value of "1" for its initial set of firmware elements 138. Accordingly, for entry 202A, the initial HFCL value in field 206A is "1". When an upgrade is made to the hardware element 128 which would make any firmware elements 138 incompatible with the

BEST AVAILABLE COPY

upgraded hardware element 128, a new row entry 202B is added for the hardware element 128. In the new row 202B, the HFCL value in field 206 is incremented and a new set of compatible firmware elements 138 is provided in another row entry in set of columns 204. For example, if hardware element 128 is a new CPU 130A requiring a new firmware element for "FirmElem2", new row 202B is added. Entry 200B provides the details of the new hardware element 128, namely "CPU 130A Rel2". HFCL field 206B is set to "2". Also, the affected firmware element 138 is updated in field 204B(2), as shown where, "FirmElem2 Rel2" replaces "FirmElem2 Rel1". If multiple firmware elements 138 are affected by the upgrade to the hardware element 128 and the members in the set of changes are compatible with each other, then the set of changes may be implemented in one new row. Otherwise, multiple firmware element 138 updates which are not compatible with each other would be made in separate new row entries, with each new row entry having the same HFCL value. However, the new row entries will have different FSCL values, described later.

[0043] When a change is made to a hardware element 128, the hardware element 128 upgrade and its effect (s) on the listed firmware elements 138 in matrix 200 must be recognized by some manner and then matrix 200 must be updated accordingly. For the present embodiment, upgrades to hardware elements 128 and their effect(s) on firmware elements 138 are tracked by design personnel responsible for designing and maintaining hardware elements 128. The personnel also update records in the SEEP, and the records of compatible elements in the software elements 136. Accordingly, the embodiment takes advantage of the previously tested and verified sets of compatabilities amongst the hardware elements 128, software elements 134 and firmware elements 138. The personnel are also responsible for making the appropriate updates to matrix 200. However, once the matrix 200 is updated, the embodiment provides an automated system which uses matrix 200 to download appropriate firmware elements 138 used with the hardware elements 128.

[0044] As with the HFCL entries 206, entries in matrix 200 indicate the compatibility levels for each upgradeable hardware element 128 vis-à-vis a particular version of a software element 134. In an embodiment a Hardware/Software Compatibility Level (HSCL) entry is provided for each upgraded hardware element 128 in matrix 200. The entry in the HSCL field 210 tracks a release level of a particular software element 134 which is compatible with that version of the hardware element 128. In the embodiment, an integer is used as a value to track release levels. A base hardware element 128 is provided with a HSCL value of "1" for its initial set of software elements. Accordingly, for entry 202A, the initial HSCL value in field 208A is "1". When an upgrade is made to the hardware element 128 which would make the particular software element 134 incompatible with the upgraded hardware element 128, a new row entry 202 is added for the hardware element 128. In the new row 202, the HSCL value in field 210 is incremented. The HSCL value is stored as a record in the software element 134, which is accessible by the firmware loader.

[0045] In another embodiment the matrix 200 is used to track hardware/software element relationships. Typically, the software elements tracked are smaller sections of executable code, like device drivers and installation code. Therein, for each hardware element 128 identified in row 202, a list of software elements 134 whose ability to function are affected by the upgrade to that hardware element 128 is provided in entries in the corresponding set of columns 208. Each entry in set of columns 208 is a reference number and version number for a particular software download for the affected software element 134. Accordingly for the example in set of columns 208, three software downloads are identified relating to the first CPU. They are identified as "SoftElem1 Rell" in column 208A(1), "SoftElem2 Rel2" in column 208A(2) and "SoftElem3 Rel1" in column 208A(3); each software download is the latest version of each software element 134 which is compatible with that upgrade version of that hardware element 128 in entry 202A. For example, if hardware element 128 is a new CPU requiring a new software element 134 for "SoftElem1", new row 202C is added. Entry 200C provides the details of the new hardware element 128, namely "CPU 130A Rel3". HSCL field 210C is set to "2". Also, the affected software element 134 is updated in field 208(1)C, as shown where, "SoftElem1 Rel2" replaces "SoftElem1 Rel1". It will be appreciated that if multiple software elements 134 are affected by the upgrade to the hardware element 128, then multiple software element 134 updates would be made in the set of columns 208. If the elements of the set of changes are compatible with each other, then one new row 202 would be added. Again, upgrades and their effects are tracked and entered manually by design personnel.

[0046] It will be appreciated that the compatibility of software elements 134 to a particular hardware element 128 is generally independent of the compatibility of that hardware element 128 to its affected firmware elements 138. Accordingly, the values in HFCL field 206 and HSCL field 210 do not necessarily correlate to each other for a given hardware element 128.

[0047] In addition to mapping current compatibilities with a hardware element 128 and a software element, a Backward Hardware/Software Compatibility Level (BHSCL) entry in field 212 provides a recordof which older software elements 134 are compatible with the current hardware element 128. When a hardware element 128 is upgraded and the previous set of software elements 134 becomes incompatible with the upgraded hardware element 128, the BHSCL value in field 212 is set to the current value of the HSCL from field 210. This indicates that there is no backward compatibility between the current hardware element 128 listed in row

BEST AVAILABLE COPY

202 with previous software elements 134. However, when an upgrade to hardware element 128 is made, causing a new row 202 to be added, and the upgraded entry in new row 202 can operate the current software elements 134, the BHSCL value in field 212 for that new row contains a list of all earlier HSCL values in fields 210 from earlier rows 202 for earlier versions of that hardware element 128 that can operate on the upgraded hardware element 128. For example, if hardware element 128 in row 202B cannot operate the software elements related to CPU version of 202A, then the BHSCL for 202C is "2". Further, if hardware element 128 in row 202C can operate the software elements related to CPU version of 202B, then the BHSCL for 202C is "2". It will be appreciated that other indices may be used in BHSCL field 212 to track compatibilities.

[0048] To use the values of the BHSCL field, the firmware loader examines the records in the software element 134 containing the list of its compatible hardware elements 128 and firmware elements 138. If the HSCL value and the FSCL value retrieved from records accessible in software element 134 do not match the values in the matrix 200, then the BHSCL and BFSCL values are accessed to locate a next best compatible set of software elements 134 and firmware elements 138. As the changes are tracked separately by the design personnel it will be assured that when they perform the updates to the SEEP 142, software element 134 records and matrix 200, a match will be provided between the given software element 134 and a set of firmware elements 138 and a hardware element 128.

[0049] For each set of firmware elements 138, a Firmware/Software Compatibility Level (FSCL) field in column 214 entry indicates the compatibility level of a software element 134 to a set of firmware elements 138. The value of an entry in column 214 is derived from the HFCL value 206 and the corresponding set of software elements 134 for a given entry in row 202. It will be appreciated that for a given HFCL value in field 206 and a given HSCL value in field 210, there may be multiple FSCL entries in column 214 wherein each entry has a row 202 in matrix 200. For each FSCL entry in column 214, its associated set of firmware elements 138 are provided in set of columns 204. Accordingly, using matrix 200 a set of firmware elements 138 can be identified which are compatible with a given software element 134. This value is also stored in a record in software element 134 which can be accessed by the firmware loader.

[0050] When an upgraded set of firmware elements 138 is not compatible with a current set of software elements 134, matrix 200 can be accessed to identify a compatible set of firmware elements 138 for the current set of software elements 134. This is accomplished by the firmware loader which accesses the FSCL values stored in the records of the software element 128 and matches it with the "best fitting" FSCL entry in matrix 200. The corresponding set of firmware elements for the row entry of the "best fitting" FSCL entry are downloaded. If there is not match, then a new application bundle 144 is required; one which contains the appropriate set of firmware elements 138 for that software element 134.

[0051] Entries in Backward Firmware/Software Compatibility Level (BFSCL) field 216 provide tracking of backward compatibility between the current set of firmware elements 138 for a hardware element 128 in a row 202 and older sets of software elements 134 associated with other rows 202. For a given hardware element 128 in a given row 202, if its associated set of firmware elements 138 is not compatible with older sets of software elements 134, then the value in BFSCL field 216 is set to the current value in the FSCL field 214, which appears in that row 202. Where the current set of firmware elements 138 is compatible with older sets of software elements 134, as well as the current set of software elements, the value of the BFSCL field 216 contains a list of all values of FSCL fields 214 that the firmware elements 138 share compatibility.

[0052] The embodiment also tracks changes made to firmware elements 138 and software elements 134 which do not affect the functionality of the existing levels of hardware elements 128. For example, a change may be made to a software element 134 to correct an internal error which affects nothing else. To track these changes, a new version of the software element 134 is used. In the new matrix version, a new version value in field 200A in entered and each entry for amended software element 134 (or firmware element 138) is updated to its new version. For example, if a software element 138 "SoftElem3" in column 208(3) is upgraded to "SoftElem3 Rev2" and the upgrade did not affect any other elements, then new software element SoftElem3 Rev2'. Also, new matrix 200' is created with identical entries to matrix 200, except that each entry for "SoftElem3" in column 208(3) is changed to SoftElem3 Rev 2. A similar revision to matrix 200 and its elements would be made for internal revisions made to firmware elements 138. It will be appreciated that historical data in older matrices 200 may be stored in archives.

[0053] For a given module 104, using matrix 200 and data on hardware elements 128 currently installed on module 104, the embodiment provides information enabling the hardware element 128 to be upgraded with its most current firmware elements 138 and software elements 134 or alternatively, to be loaded with a compatible set of firmware elements 138 and software elements 134. The embodiment allows the firmware elements 138 and software elements 134 to be upgraded independently of each other.

[0054] Referring to Fig. 1, as indicated earlier, for a given hardware element 128, numerous sets of differing firmware elements 138 and software elements 134 can be associated with it, providing current and backward operational compatibility. In order to track the sets of firmware and software element downloads, the embodiment groups together a set of firmware elements 138

BEST AVAILABLE COPY

with a set of compatible software elements 134 into an application bundle 146. A group of application bundles 146 may be stored at a central location, such as a server or an FTP site, or they may be stored on a CD ROM. In the embodiment, application bundles 146 are stored in flash banks 148 or some non-volatile memory located either on the card 100 or in an external module or SEEPs 142 of memory in the main controller 102. Once an appropriate application bundle 146 is identified for a given hardware element 128, the application bundle 146 can be downloaded into the programming system relating to the hardware element 128 and the associated devices for the firmware and software elements can be installed onto the local firmware and software elements. A unique identifier associated with each application bundle 146, in order to track the bundles 146.

[0055] To download an appropriate application bundle 146, the embodiment needs to identify the target application bundle 146 containing the target software element 134 and the target firmware element 138. The flash banks 148 provide a main and an alternate set of storage devices allowing a downloaded application bundle 146 to be tested and configured on the alternate set of storage devices before switching over operation of the system to the downloaded set of firmware and software elements on the next reset of the system.

[0056] In order to identity the appropriate configuration for a given hardware element 128 in a given hardware module 104 for the application bundle 146, it is necessary to have information regarding the configuration of the target hardware element 128. Accordingly, configuration information about hardware elements 128 for a given module 104 is stored in SEEP 144. The SEEP stores data from the current HFCL field 206 and HSCL field 210 for the hardware element 128, thereby providing information on firmware and software compatibilities with the present hardware elements 128.

[0057] Further detail on the application bundle 146 is provided. Application bundle 146 comprises one software element 134, a firmware loader and a set of all possible firmware element 138 loads for that software element 134.

[0058] The firmware loader is the interface between the application bundle 146 and the hardware element 128. The firmware loader is a software program which loads the firmware elements 138 into their respective memory devices 142. Upon activation, the firmware loader examines the configuration information for the hardware elements 128 on each card. Accordingly, the firmware loader obtains the values of the HFCLs for each hardware element 128 of each module 104. In order to identify the current set of firmware elements 138 which are the most current set for the hardware element 128, the loader utilizes the HFCL value for the hardware element 128, which is stored locally on module 104 in its SEEP 144. As described above, the HFCL value provides an ordinal marker to a level of firmware elements 138 for a hardware element 128. In order to provide the list of firmware elements 138 relating to the HFCL value, the application bundle 146 has access to a look-up table comprising a list of HFCL values for that hardware element 128 and for each value, a corresponding list of firmware element downloads. This look-up table is derived from value in columns 202, 206 and 204 of matrix 200. If a different version of a software element exists, then another application bundle 146 is made for it.

[0059] Based on the information in SEEP 144, the loader accesses matrix 200 and determines whether or not the hardware element 128 has the proper firmware and software elements. For firmware element 138 upgrades, if the proper firmware element 138 loads are available, the loader will transfer all the downloads for the firmware elements 138 to their devices. If the proper firmware elements 138 are not available, an error message is generated. For example, an error would be generated if the loader does not have any entries in its matrix 200 for the HFCL for a given hardware module 104. In this case, the download would fail and the application bundle 146 would be discarded.

[0060] With the matrix 200, the application bundle 146 and the data in SEEP 144, the embodiment can download firmware and software elements into a module 104. Following is a description of a downloading method used by the embodiment.

[0061] Referring to Fig. 3, controller 302 is shown connected via bus 300 to modules 304a and 304b. Bus 300 may comprise an Ethernet link, an optical link or other communication technologies known in the art. In the same manner as in the circuit card 100 where main controller 302 is able to download sets of firmware and software elements to each of its modules 104, main controller 302 can download to its modules 304 over bus 300 the correct download information. It will be appreciated that information from each local SEEP must be transmitted via the protocol used by bus 300 to the main controller. And main controller 302 must transmit the appropriate application bundle 146 over bus 300 to each module 304a and b.

[0062] Referring to Figs. 1 and 4, following is an example of an illustrative download system 400 for an embodiment. Therein, resource module 402 is connected via a communication link 404 to main controller 104. Resource module 402 comprises an application bundle library 406, which contain a series of application bundles 146A ... 146N for many hardware elements 128, as described earlier. Each application bundle 146 has one software element 134 associated with a set of firmware elements 138. The same software element 134 may be associated with several different sets of firmware elements 138 in different bundles. The operator of main controller 104 selects the resource module 402 from which the firmware elements 138 and/or software elements 134 are to be downloaded. As described earlier resource module 402 may be a CD-ROM, and FTP site, a web server or other known data sources. Next, the operator selects the desired application bundle 146 from

BEST AVAILABLE COPY

the resource module 402. This is controlled by download software which can interface with the operator and module 104. Next, one of the flash banks 148 in module 104 which is to receive the application bundle 146 is selected. Generally, the flash bank 148 which is selected is the non-operational flash bank, i.e. the current firmware and software elements are provided from the current flash bank 148. The new application bundle is downloaded to the inactive flash bank.

[0063] After the download of the application bundle 146 is completed to the selected flash bank 148, it is verified using a checksum associated with the application bundle 146 and validation of the application bundle identification number. Second, the firmware loader and its loads are verified. This verification process may simulate or even operate the firmware loader to determine if the loader can generate proper loads for the firmware build without error.

[0064] As noted earlier, the firmware loader can access the values of the HFCLs in SEEP 144 and utilizes this information and its internal matrix 200 to determine whether the appropriate firmware element 138 downloads are installed on the module 104. If not, the download in the bank 148 is erased.

[0065] If the loader recognizes all the cards and their HFCLs and has the appropriate loads for the cards and recognizes all of the cards HSCL and can properly process them and recognize the FSCLs generated for this hardware and can properly process them the verification process is complete and successful.

[0066] Upon successful completion of the download, the operator selects the flash bank 148 to be made active and the new application bundle 146 is activated.

[0067] Volatile devices are downloaded with their program elements. These devices must have their downloads reinstalled every time card 100 is reset. The firmware elements 138 for the non-volatile devices are loaded by the firmware loader from a new bank 120 when the firmware loader determines that the firmware element must be upgraded. In the embodiment, the firmware elements 138 are derived from JEDEC files.

[0068] After all firmware elements 138 have been downloaded, the firmware loader processes the software element 134 download. The software element 134 download may be an upgraded version of the software element 134. The software element 134 can read and use values of the HSCL and FSCL in SEEP 144. For example, after obtaining the HSCL and FSCL values, the software element 134 may select one of several predefined values for variables for an equation. This may occur when the equation is based on values produced from a particular upgradeable hardware element 128, such as a resistor. For the given hardware element 128, the software element 134 requires the values of the HSCL and the FSCL in order to determine the compatibility of the software element 134 with the existing other elements. The firmware loader queries the software element load for a list of all HSCLs that the software element 134 recognizes. From the information stored in the matrix 200, the firmware loader can determine which of the HSCLs to return to the software element 134 for this module 104. This is done by analyzing the BHSCL value stored in the matrix 200. If the change to hardware element 128 does not create an incompatibility issue with the old software element 134, then the old software element 134 is allowed to continue to function. This interface is required, as if a new HSCL value were provided directly to old software element 134, then the old software element 134 would not recognize the new HSCL. Therefore the software element 134 would halt and advise the operator either to upgrade the software element (or downgrade the hardware element 128). Where an old software element 134 may function properly with a new hardware element 128, then only a new loader is issued which translates the new HSCL stored in the new hardware element 128 into a value that the old software element 134 recognizes.

[0069] As noted above, the BHSCL contains a list of all HSCL values with which the current HSCL is compatible. If the BHSCL does not have a value which equals the HSCL for a given module and a value of HSCL, then there is backward compatibility. For example, if the value in the SEEP were 5 and 3, then according to the table an HSCL having a va lue of 5 is compatible with and HSCL of 4 and 2, but not 1, 3 or 0. The values of 5, 4 and 2 are then compared to the values passed to the firmware loader by the software element 134 load. The most significant matching value, if any, would be used to pass back to the software element as the HSCL value.

[0070] This process is repeated until all module levels have been resolved and an HSCL value has been passed to the software element for each module. Similarly, the same process is repeated to establish an acceptable FSCL for each software for each module.

[0071] For a given HFCL and HSCL there may be multiple FSCL entries each having a new row in the matrix 200. Each of the multiple FSCL values would have its own row in the matrix 200 and could have its own set of routines within the software element 134 associated with the application bundle 146. This would allow a software element to negotiate a compatible set of firmware element 138 loads. This would allow firmware element changes that are incompatible with an old software element 134 to be bypassed, allowing old software element 134 to still function, albeit without the latest set of firmware element 138 versions.

[0072] It will be appreciated that the embodiment provides the ability of determining whether a software element 134 is compatible with an installed target hardware element 128. This is achieved by accessing and comparing the values in the HSCL and HFCL values stored in the SEEP 144 associated with the target hardware element 128. The values in the SEEP 144 are compared against the values in the matrix 200 in the application bundle 146 associated with the particular software ele-

BEST AVAILABLE COPY

ment 134. If the comparison indicates that there is a match, then the software element 134 is compatible with the target hardware element 128. If the comparison indicates that there is no match, the embodiment may notify the operator of the switch of the mismatch via an appropriate error message.

[0073] Referring to Fig. 5, another embodiment is shown relating to network 500. Network 500 comprises a series of communication elements 502 which are connected together via links 504. A network station 506 provides overall control and configuration of communication elements 502 and thereby the operation of network 500.

[0074] Communication elements 502 comprise various hardware, software and firmware elements. Network station 506 contains resource module 402. Resource module 402 contains the application bundle library 406 which contains the application bundles 146A...146N. Downloads of application bundles 146 to any of communication elements 502 are effected from network station 506 by software operating thereon following the procedure described above for Figs. 1-4. It will be appreciated that other download methods may also be used.

[0075] From the above description, it will be appreciated that the embodiment provides a centralized system for tracking independently compatibilities for firmware and software elements which are affected or affect the operation of hardware elements in electronic circuits. The tracking system is also used as a component in a downloading system for the firmware and software elements.

[0076] It will further be appreciated that in other embodiments, the system and method of tracking firmware elements 138 and software elements 134 may be reversed from the embodiment described above.

[0077] It is noted that those skilled in the art will appreciate that various modifications of detail may be made to the present embodiment, all of which would come within the scope of the invention.

Claims

1. Method of selecting a version of a program code from a plurality of program codes for a program storage element associated with a hardware element for a circuit board, said hardware element being implementable in a plurality of hardware element versions, said method comprising:

   identifying a particular program code of said plurality of program codes which is compatible with a particular hardware element of said plurality of hardware element versions on said circuit board;

   selecting from a group of program codes selected from said plurality of program codes a selected program code matching said particular program code; and

   loading said program storage element with said selected program code.

2. Method of selecting a version of a program code from a plurality of program codes for a program storage element as claimed in claim 1 wherein

   information regarding said particular program code is stored with said circuit board; said plurality of program codes are stored in a program code bundle; and

   said selected program code is identified by accessing a list, said list comprising one entry for each of said plurality of hardware element versions, each entry in said list having a reference to a compatible program code of said plurality of program codes which can operate therewith.

3. Method of selecting a version of a program code from a plurality of program codes for a program storage element as claimed in claim 1 or claim 2 wherein each of said plurality of program codes is a firmware program.

4. Method of selecting a version of a program code from a plurality of program codes for a program storage element as claimed in claim 2 or claim 3 wherein

   a software program is associated with said program code bundle, said software program being associated with said hardware element.

5. Method of selecting a version of a program code from a plurality of program codes for a program storage element as claimed in any one of claims 2 to 4 wherein

   said each entry in said list is further associated with a list of earlier versions of each program code of said plurality of program codes with which said hardware element is compatible.

6. Method of selecting a version of a program code from a plurality of program codes for a program storage element as claimed in any one of claims 2 to 5 wherein

   said each entry in said list is further associated with a list of earlier versions of said software program with which said hardware element is compatible.

7. System for processing program updates associated with a hardware element being implementable in a plurality of hardware element versions in a circuit board, said system comprising:

   a record of said plurality of hardware element versions, each entry in said record being asso-

BEST AVAILABLE COPY

ciated with a compatible program code from a plurality of program codes;
a second record associated with said circuit board, said second record identifying a particular program code of said plurality of program codes which is compatible with an installed hardware element version of said hardware element on said circuit board;
a bundle of executeable program codes comprising a group of said plurality of program codes; and
a program code loader adapted to access said record and said second record to identify a suitable program code from said plurality of program codes for said installed hardware element version, adapted to locate said suitable program code in said bundle, and adapted to effect transfer of said suitable program code from said bundle to a storage device adapted to receive said suitable code, said storage device associated with said installed hardware element version and said circuit board.

8. System for processing program updates as claimed in claim 7 wherein each of said plurality of program codes is a firmware program.

9. System for processing program updates as claimed in claim 7 or claim 8 wherein
a software program is associated with said program code bundle, said software program associated with said hardware element.

10. System for processing program updates as claimed in any one of claims 7 to 9 wherein each said entry in said record is further associated with a compatible software program therewith.

11. System for processing program updates as claimed in claim 10 wherein
said each entry in said record is further associated with a list of earlier versions of each program element of said plurality of program elements with which said hardware element is compatible.

12. System for processing program updates as claimed in claim 10 or claim 11 wherein
said each entry in said record is further associated with a list of earlier versions of said software program with which said hardware element is compatible.

13. Method for updating a program code from a plurality of program codes for a hardware element being implementable in a plurality of hardware element versions in a circuit board, said method comprising:

identifying a particular program code of said plurality of program codes which is compatible with a particular hardware element of said plurality of hardware element versions on said circuit board;
selecting from a group of program codes selected from said plurality of program codes a selected program code matching said particular program code; and
transferring said suitable program code from said group of program codes to a storage device adapted to receive said suitable code, said storage device associated with said particular hardware element and said circuit board.

14. Method for updating a program code from a plurality of program codes for a hardware element as claimed in claim 13 wherein
information regarding said particular program code is stored with said circuit board;
said plurality of program codes are stored in a program code bundle; and
said selected program code is identified by accessing a list, said list comprising one entry for each of said plurality of hardware element versions, each entry in said list being associated with a reference to a compatible program code of said plurality of program codes which can operate therewith.

15. Method for updating a program code from a plurality of program codes for a hardware element as claimed in claim 14 wherein each of said plurality of program codes is a firmware program.

16. Method for updating a program code from a plurality of program codes for a hardware element as claimed in claim 14 or claim 15 wherein a software program is associated with said program code bundle, said software program associated with said hardware element.

17. Method for updating a program code from a plurality of program codes for a hardware element as claimed in any one of claims 14 to 16 wherein said each entry in said list is further associated with a list of earlier versions of each program code of said plurality of program codes with which said hardware element is compatible.

18. Method for updating a program code from a plurality of program codes for a hardware element as claimed in any one of claims 14 to 17 wherein said each entry in said list is further associated with a list of earlier versions of said software program with which said hardware element is compatible.

19. Method for updating a program code from a plurality of program codes for a hardware element as claimed in any one of claims 13 to 18 wherein said

BEST AVAILABLE COPY

storage device has an active bank and an inactive bank and said suitable program code is transferred to said inactive bank.

20. Method for updating a program code from a plurality of program codes for a hardware element as claimed in claim 19 wherein said suitable program code is verified for integrity and upon verification of same, said storage device switches from said inactive bank to said active bank.

21. Method of selecting a version of a firmware code from a plurality of firmware codes for a firmware element associated with a hardware element for a circuit board said hardware element being implementable in a plurality of hardware element versions, said hardware element being further associated with a software element, said method comprising:

identifying a particular version of said pluarlity of hardware element versions for said hardware element which is associated with said circuit board;
identifying a particular firmware code of said plurality of firmware codes which is compatible with a particular hardware element of said plurality of hardware element versions on said circuit board;
selecting from a group of firmware codes selected from said plurality of firmware codes a selected firmware code matching said particular firmware code;
associating a software code with said group of firmware codes;
loading said firmware element with said selected firmware code; and
checking compatibility of operation of said software code with said selected firmware code and said particular version of said plurality of hardware element versions,

wherein

said selecting from a group of firmware codes from said plurality of firmware codes a selected firmware code is done independently of installation of said software code.

22. Method of evaluating compatibility of a hardware element in a circuit card with a program code, said hardware element being implementable in a plurality of hardware element versions, said program code being implementable in a plurality of program code versions, said method comprising:

identifying a particular version of said plurality of hardware element versions for said hardware element which is associated with said circuit board;
identifying a particular program code of said plurality of firmware codes which is associated with said hardware element;
evaluating whether said particular program code is compatible with said particular version of said plurality of hardware element versions; and
indicating whether said particular program code is compatible with said particular version of said plurality of hardware element versions.

23. Method of evaluating compatibility of a hardware element in a circuit card with a program code as claimed in claim 22 wherein

information regarding said particular program code is identified by accessing a list, said list comprising one entry for each of said plurality of hardware element versions, each entry in said list having a reference to a compatible program code of said plurality of program codes which can operate therewith.

24. Method of evaluating compatibility of a hardware element in a circuit card with a program code as claimed in claim 22 or claim 23 wherein each of said plurality of program codes is a firmware program.

BEST AVAILABLE COPY

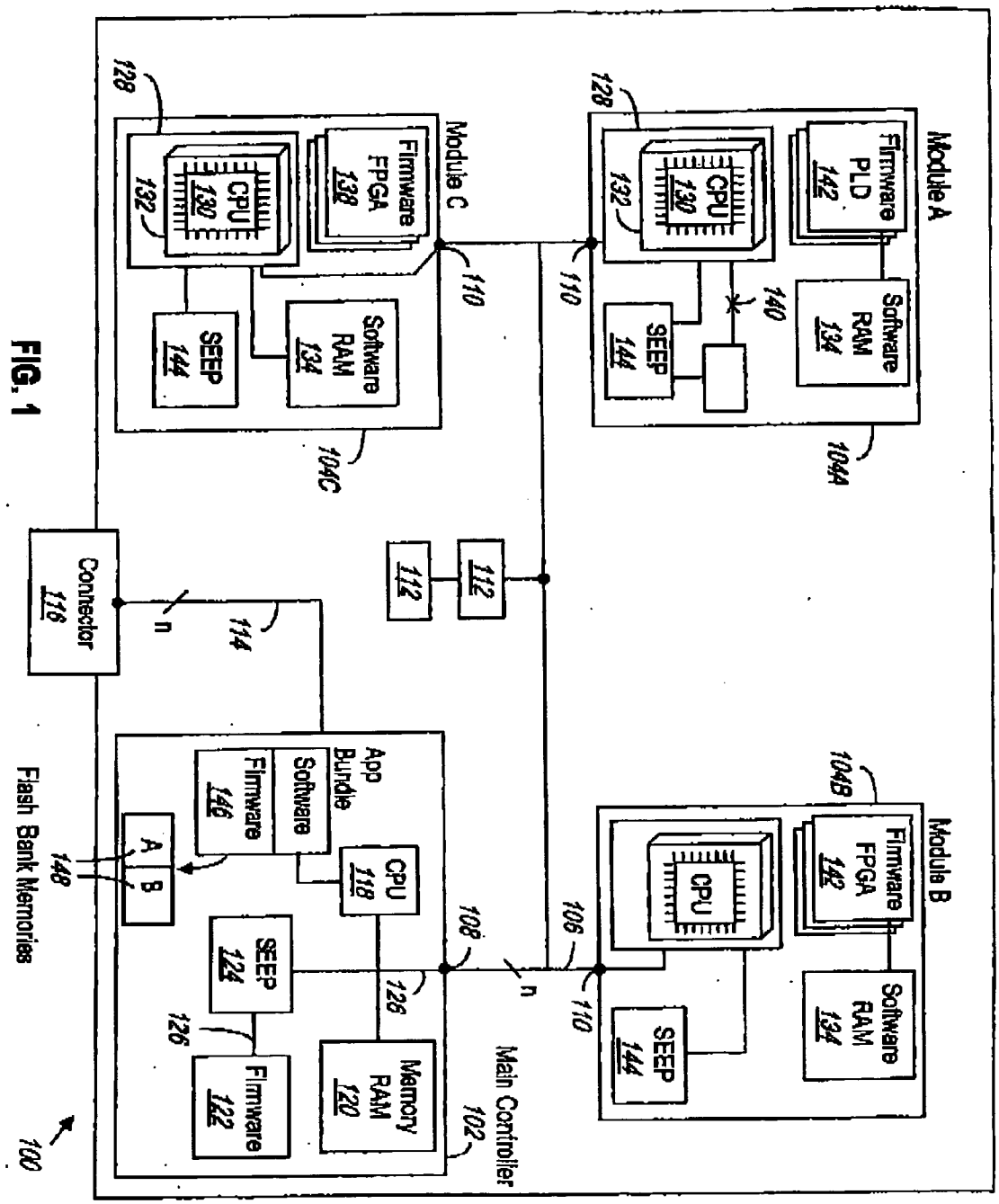

FIG. 1

BEST AVAILABLE COPY

BEST AVAILABLE COPY

MAR-15-2006 12:56 HP LEGAL FT COLLINS 9708987247 P.19

EP 1 335 283 A2

	Module	HFCL	HSCL	BHSCL	FSCL	BFSCL	Firmware ID	Firmware ID	Firmware ID	Software ID	Software ID	Software ID
202A	CPU 130A Rel 1	1 (206A)	1				FirmElem1 Rel 1	FirmElem2 Rel 1	FirmElem3 Rel 1	SoftElem1 Rel 1	SoftElem2 Rel 1	SoftElem3 Rel 1
202B	CPU 130A Rel 2			2	1 (206B)	2	FirmElem1 Rel 1	FirmElem2 Rel 2 (204(2)B)	FirmElem3 Rel 1			
202C	CPU 130A Rel 2		2 (210C)		2					SoftElem1 Rel 2 (208(1)C)	SoftElem2 Rel 1	SoftElem3 Rel 1
202D	CPU 130B Rel 1											
202E	Board Rel 1											
	200	*206*	*210*	*212*	*214*	*216*	*204(1)*	*204(2)*	*204(3)*	*208(1)*	*208(2)*	*208(3)*

Card 100 Rel 1 – 200A

FIG. 2

PAGE 19/28 * RCVD AT 3/15/2006 2:48:42 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-6/35 * DNIS:2738300 * CSID:9708987247 * DURATION (mm-ss):08-46

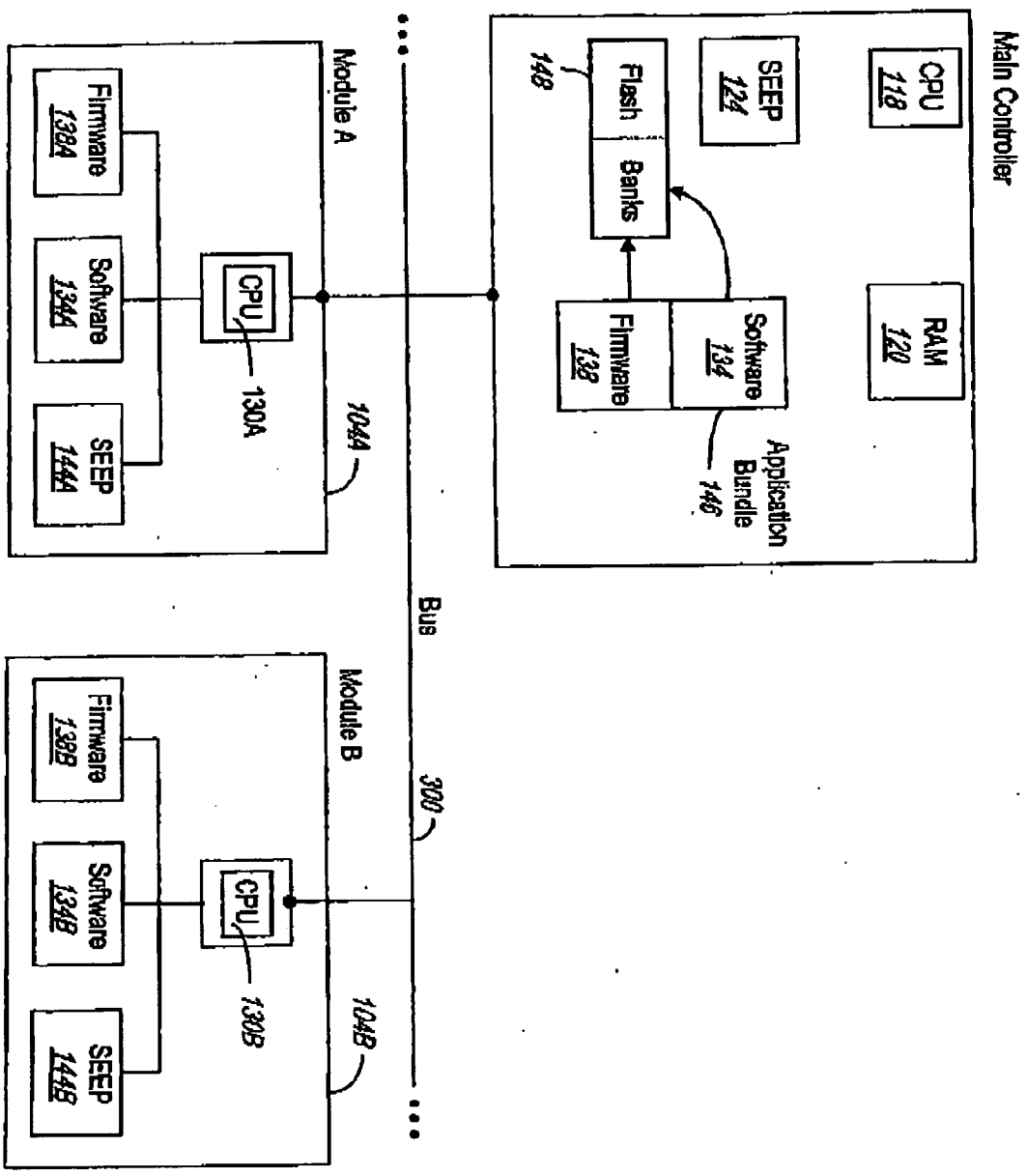

FIG. 3

BEST AVAILABLE COPY

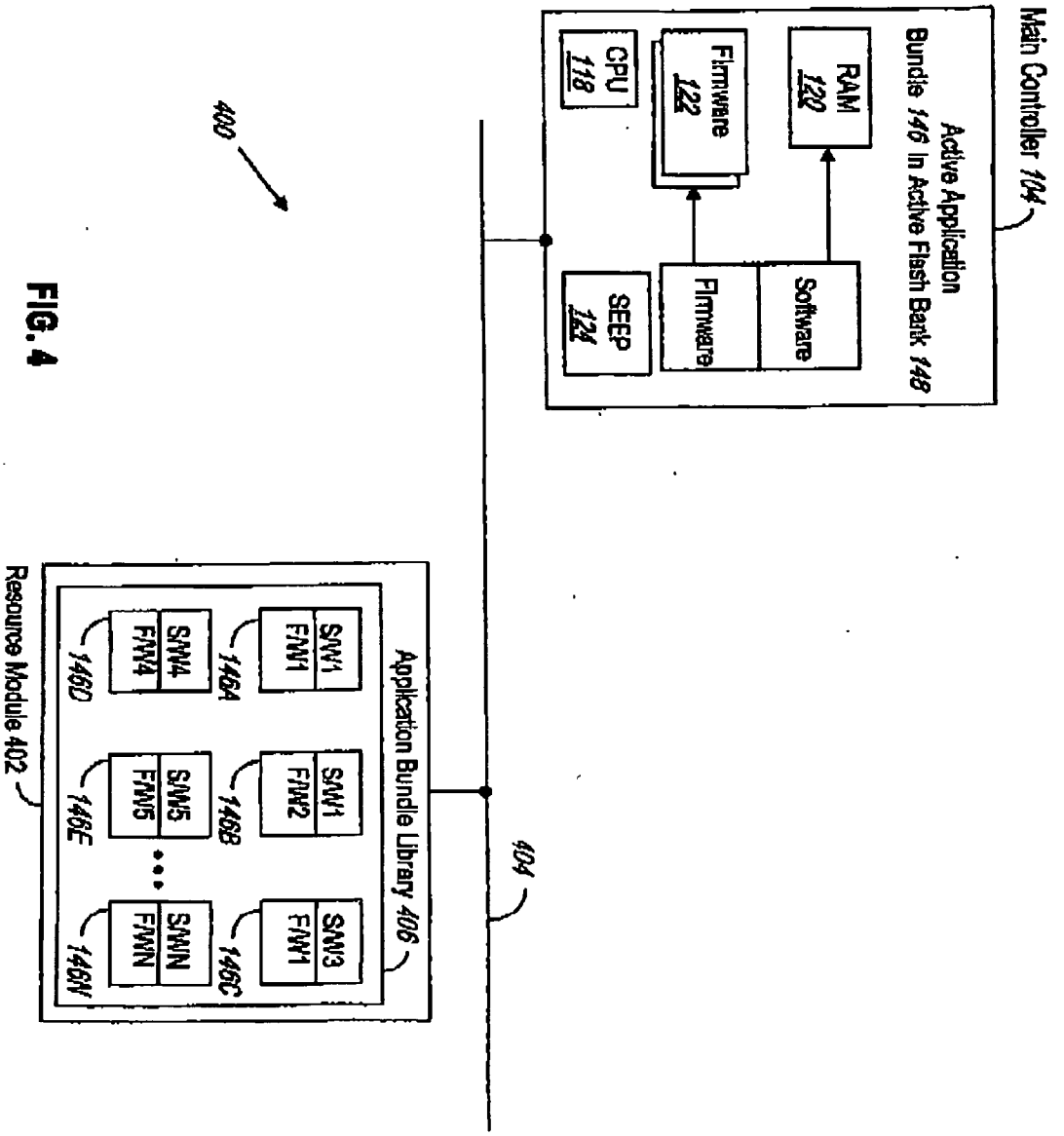

FIG. 4

BEST AVAILABLE COPY

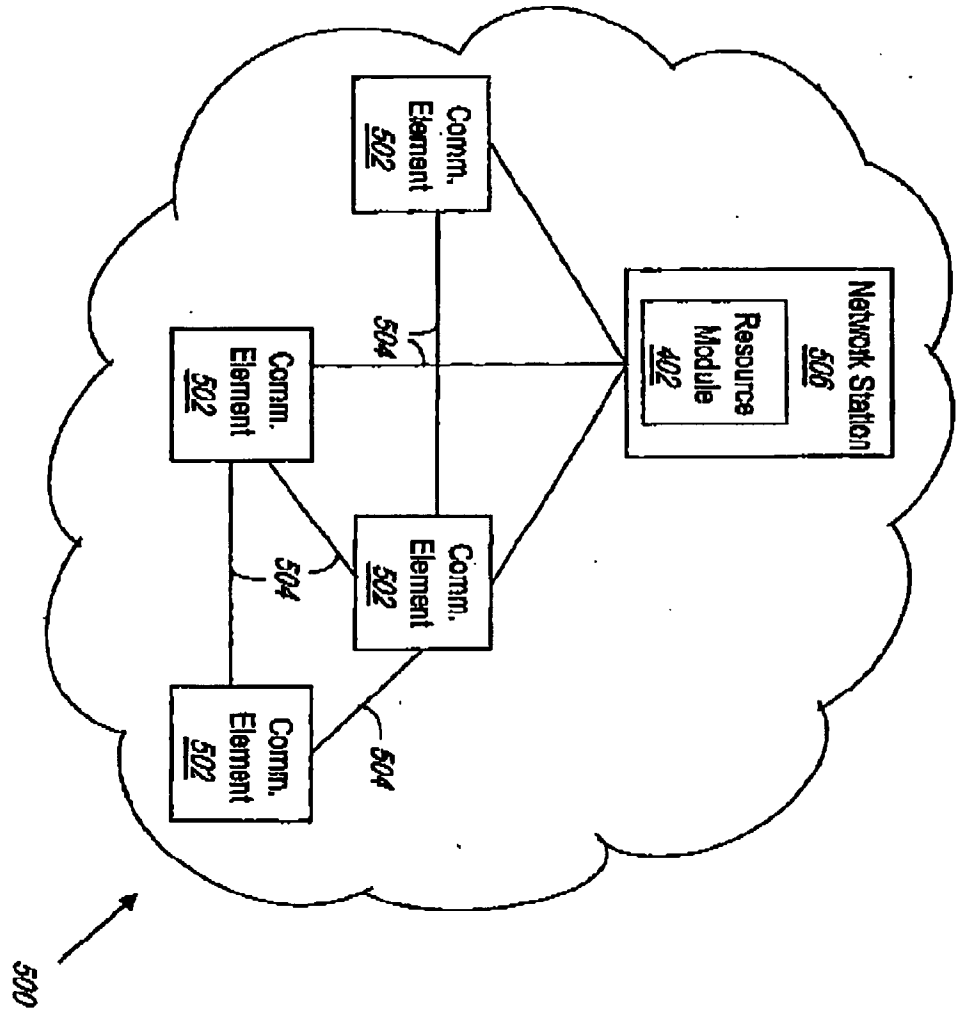

FIG. 5

BEST AVAILABLE COPY